

Summary

Builder and leader. Has *broad* knowledge of modern backend technologies and people. Has *deep* understanding of architecture, high performance and security.

In other words, he's productive at being a senior backend developer, architect, SRE, or tech lead.

Experience

- since 2020 - Huawei, [Cloud](#) business unit. Principal engineer.
Working on security *inside* the cloud. Team size: 7.
 - First engineer on a team. Interviewed 30+ candidates.
 - Learned what Chinese colleagues have, which processes they follow, which tools they use.
 - Gathered requirements to SOAR from zero, developed domain model. [My explanation](#) of what SOAR is.
 - Architected SOAR for internal (Huawei security operations center) and external (cloud customers) users.
 - Established development workflows and rules.
 - Deployed Kubernetes cluster for the team (using [Kubespray](#)). Connected it to Gitlab CI.
 - Built playbook subsystem consisting of 5 microservices.
 - Represented the team in roadmap negotiations with Chinese customers and teams.
 - Buzzwords: DDD, Java, Spring + Spring Boot, Spock, REST, OpenAPI/AsyncAPI, Kubernetes, Gitlab CI, Helm, [Jib](#), Openwhisk, Flowable (BPMN engine), PostgreSQL, Apache Kafka, ANTLR.
- 2019 – 2020 - [Positive Technologies](#). Senior programmer.
Cybersecurity company. Working on security threat gathering and filtering ([threat intelligence platform](#)). Team size: 15.
 - Integrated third-party threat intelligence feeds.
 - Fixed a huge chunk of technical debt. Increased write/read throughput by two orders of magnitude.
 - Helped the team to release the platform on schedule.
 - Buzzwords: DDD, Go, REST, go-chi, Python 3, pytest, Ansible, docker-compose, MinIO, PostgreSQL.
- 2017 – 2020 – [Novosibirsk State University](#). Assistant professor, invited expert.
 - Taught 4 separate courses on [operating systems](#), [parallel programming](#), computing history, [security](#).
- 2015 – 2019 – [2GIS](#). Senior programmer.
The company does web and mobile applications with directory, map and navigator. Working on backend projects, supporting the applications. Team size: 10.
 - Maintained a content delivery network. It was providing map data to 20 millions of users. Automated operator workflow so updated maps could be delivered in one click daily. The updates were monthly before that. Improved Python code quality and performance a lot. The service could churn a task for a week, the run time was brought to two hours.
 - Built a fault-tolerant cross datacenter pipeline based on Apache Kafka. It replaced an old single datacenter ZeroMQ-based pipeline. Made the transition seamless for consumers. The pipeline was used to deliver business statistics from users.
 - Brought a new life to a dying Python-based geopositioning service. It was slow, hard to scale, and fault intolerant. Rewrote and stabilized it using asyncio. As the load increased with years, rewrote it in Go. Now it handles heavy load on commodity hardware.
 - Created and executed several long-term team roadmaps. Decomposed, estimated and prioritized hundreds of tasks. Actively participated in sprint planning and process improvement. Coordinated cross-team feature development.
 - Buzzwords: Python 3, aiohttp, Falcon, mypy, pytest, SQLAlchemy, SQLite, Go, C++14, rdafka, RapidJSON, Scala, ScalaTest, Vue.js, Apache Kafka, ZeroMQ, Apache Thrift, MongoDB, Redis, Tarantool, Zookeeper, PostgreSQL, ClickHouse, ElasticSearch, syslog, apt, LXD, Docker, Ansible, Jenkins, Grafana, Zabbix.

Education

- 2009 – 2015. [Novosibirsk State University](#). Master's degree in Information Technology. Specialized in [parallel computing](#) and systems programming.

Languages

Doesn't care about languages much (see buzzword zoo above), but knows some of them well enough to be productive without warmup.

- *Python*. 8 years of experience, daily use for 4 years. Type-hinted an entire codebase to increase maintainability and to find hidden bugs (found dozens). Can explain and use [import this](#). [Contributed](#) to mypy.
- *Go*. 4 years of experience, daily use for 1.5 years. Wrote a dozen of services from scratch, rewrote a couple of Python services in Go (it was very satisfying) and supported huge monolithic service. Wrote a couple of blog posts about Go, for example, [a guide to structured logging in Go](#) and [a note on database/sql connection pools](#).
- *Java*. Daily use for the last year. Cannot recommend it to anyone, most of the ecosystem is broken by design. It can be tolerated if used with care, though.

Skills and interests

- *Learning*. Lifelong learner. There's a [bookshelf](#).
- *GNU/Linux*. Built his own Linux from scratch (at home). Knows enough about UNIX-like OSs to teach it at the university.
- *Distributed systems*. Defended a master's thesis on this topic. Appreciates the motto "Reliability, Maintainability, Scalability!". Can explain what's happening in tcpdump + wireshark, Zookeeper and WALs.
- *UX*. Cares about UX even though it's "not backend responsibility".
- *Architecture*. A proponent of Clean architecture and domain-driven development. Knows how to use them in practice. Understands how to model complex domains.
- *Documentation*. Knows how to document architecture, APIs, SLAs, postmortems, tutorials and so on.
- *Communication*. Knows how to find real problems in generic complaints. Knows common cognitive biases. Knows how to lead and be led. Understands that programming is not just about writing code. The code ends up being less important than the architecture, and the architecture ends up being less important than social issues.

Wants

- Working with people better than him. Domain experts – awesome. Technical experts – good.
- Productivity tools. Modern CI/CD or green light to build one. Modern tools for storing non-code artifacts (documentation, tasks, test cases and so on).
- Distributed systems under load. A connection with security field is a plus.
- Technical track growth. The starting point is Staff software engineer, ideal – Principal engineer. Titles here use [Mozilla's interpretation](#).